IN THE CLAIMS:

Claims 1-5 having previously been cancelled, please cancel claims 6 through 16, without prejudice.

Please add new claims 17 through 29 as follows:

--17. (New) A method of symmetric multiprocessing for an inter-process control (IPC) message-passing operating system where operating system calls execute in critical and non-critical areas, said method comprising the steps of:

responding to an operating system call requiring access to a critical area of said IPC message-passing operating system by:

requesting a global lock; and

responding to said global lock being available by performing the steps of:

acquiring said global lock;

performing said operating system call in said critical area of said IPC message-passing operating system; and

releasing said global lock; and

responding to said operating system call requiring access to a non-critical area of said IPC message-passing operating system by:

performing said operating system call in said non-critical area of said IPC message-passing operating system.

18. (New) A method as claimed in claim 17, wherein said IPC message-passing operating system includes a micro kernel operating system and wherein:

said step of responding to an operating system call requiring access to a critical area of said IPC message-passing operating system, includes the step of responding to an operating system call requiring access to a critical area of said micro kernel operating system;

said step of performing said operating system call in said critical area of said IPC message-passing operating system, includes the step of performing said operating system call in said critical area of said micro kernel operating system;

said step of responding to said operating system call requiring access to a non-critical area of said IPC message-passing operating system includes the step of responding to said operating system call requiring access to a non-critical area of said micro kernel operating system; and

said step of performing said operating system call in said non-critical area of said IPC message-passing operating system includes the step of performing said operating system call in said non-critical area of said micro kernel operating system.

19. (New) The method as claimed in claim 18, wherein said micro kernel operating system includes a pre-emptable micro kernel operating system, said method further comprising the steps of:

pre-empting any non-critical threads currently executing on said pre-emptable micro kernel operating system prior to said step of acquiring said global lock; and

reinstating said pre-empted threads following said step of releasing said global lock.

20. (New) The method as claimed in claim 19, wherein said step of performing said operating system call to said critical area comprises the steps of:

locking said critical area of said pre-emptable micro kernel operating system;

entering said critical area of said pre-emptable micro kernel operating system; executing operating system functions as required; and

exiting said critical area of said pre-emptable micro kernel operating system.

21. (New) The method as claimed in claim 20, further comprising the step of prioritizing execution of threads in accordance with how their respective call latencies will impact real time operation.

22. (New) The method as claimed in claim 20, wherein said operating system includes a real time operating system, and said method further comprises the step of scheduling execution of said threads to be performed by predetermined time deadlines.

23. (New) A computer system comprising:

one or more processors;

a memory medium storing an inter-process control (IPC) message-passing operating system where operating system calls execute in critical and non-critical areas, in a machine executable form, and a lock manager in a machine executable form;

a communication network interconnecting said one or more processors, and said memory; and

said lock manager being operable to:

respond to an operating system call requiring access to a critical area of said IPC message-passing operating system by:

requesting a global lock; and

responding to said global lock being available by performing the steps of:

acquiring said global lock;

performing said operating system call in said critical area of said IPC message-passing operating system; and

releasing said global lock.

24. (New) An apparatus for symmetric multiprocessing comprising:

an inter-process control (IPC) message-passing operating system means where operating system calls execute in critical and non-critical areas;

means responsive to an operating system call requiring access to a critical area of said IPC message-passing operating system by:

requesting a global lock; and

responding to said global lock being available by performing the steps of:

acquiring said global lock;

performing said operating system call in said critical area of said IPC message-passing operating system; and

releasing said global lock.

25. (New) A computer readable memory medium, storing computer software code executable to perform the steps of:

responding to an operating system call requiring access to a critical area of an IPC message-passing operating system by:

requesting a global lock; and

responding to said global lock being available by performing the steps of:

acquiring said global lock;

performing said operating system call in said critical area of said IPC message-passing operating system; and releasing said global lock.

26. (New) A computer data signal embodied in a carrier wave, said computer data signal comprising a set of machine executable code being executable by a computer to perform the steps of:

responding to an operating system call requiring access to a critical area of an IPC message-passing operating system by:

requesting a global lock; and

responding to said global lock being available by performing the steps of:

acquiring said global lock;

performing said operating system call in said critical area of said IPC message-passing operating system; and

releasing said global lock.

27. (New) The method as claimed in claim 17, where said critical area of said IPC message-passing operating system is limited to the message passing functionality of said IPC message-passing operating system, and wherein said step of performing said operating system call in said critical area of said operating system comprises the step of:

performing an IPC message-pass operation for said operating system call.

28. The method as claimed in claim 27, where said IPC message passing operating system requires a message-pass before and after execution of said operating system call in said non-critical area of said IPC message-passing operating system, said method comprising the subsequent steps of:

requesting a global lock a second time; and

responding to said global lock being available by performing the steps of:

acquiring said global lock a second time;

performing a second message-pass operation for said IPC message-passing operating system call; and

releasing said global lock a second time.

29. (New) The method as claimed in claim 17, wherein said IPC message-passing operating system includes a micro kernel operating system having operating system calls executing in external processes, and wherein said step of performing said operating system call in said non-critical area of said IPC message-passing operating system comprises the step of:

performing said external process for said operating system call.--